PACMAN VS. GHOSTS

PROGRAMMING ASSIGNMENT 1 ~ CS/IMGD 4100 ~ 2017 B TERM
DEADLINE: NOVEMBER 3, 11:59PM

DESCRIPTION

For this assignment, you will be creating AI to control the ghosts in Pacman. There are two main phases to this assignment:

1. Creating an AI controller for the ghosts, and
2. Playtesting the AI controller with your friends.

You will be turning in:

1. Source code for your assignment.
2. A short assignment writeup.
3. A readme text file containing: your name, the names of any people who helped you with the assignment, references to any external sources you used, and any special instructions necessary for getting your code to work.

PHASE 1: CREATING GHOST AI

We will be using the Pacman vs. Ghosts framework for this assignment, which has historically been used in game AI research competitions. It is written in Java, and has been tested and confirmed to work with JDK 9. The framework is available for download from the course website.

Pacman vs. Ghosts provides a clean interface for you to write a Java agent to control the ghosts, and also comes with several example agents for both Pacman and the ghosts (including a pass-through controller that permits a human to control Pacman). It also provides some important pathfinding functions that you are permitted to use in the assignment. Note that an important part of this class is becoming comfortable with using other engines and frameworks; it is your responsibility to read example code and learn how this framework works. Feel free to work together to understand the framework, and ask for help on Slack.

It is not necessary to alter the framework in any way, other than changing a line in Executor.java that determines which controllers will be used in the game. You should write your controller as a single class in the *pacman.entries.ghosts* package. Make sure your name is in the comments at the top of the file. The only code you will turn in is the .java file for your new controller class.

MIMICKING TRADITIONAL GHOST AI

You will be implementing a controller that mimics the original ghost behavior for Pacman. You can find information about how the original "AI" worked in several places online; one of these sources is here:

http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior

Note that some sources use different names for the ghosts than the framework uses.

Because of the way the framework is set up, it may not be possible to *perfectly* replicate the original AI design. Try to come as close to the original as possible, and note (briefly) in your writeup any places that you deviate from the original behavior due to framework limitations. In particular, the *scatter* behavior is not possible to perfectly replicate using the Pacman vs. Ghosts framework.

It is recommended that you begin by implementing a simple AI that has the ghosts move entirely randomly, and then build up your controller from there. Partial credit can only be given if your code is readable, compiles, and executes, so make sure that you always have something that works that you can turn in!

PHASE 2: PLAYTESTING

You will be conducting a comparative playtest to see how your controller fares against a baseline "random" AI. There is a built-in ghost AI controller you can use for this. This will be a small playtest, to give you a sense of how they are run in the industry. Note that the results you get from a very small population playing your game are not enough to come to a scientifically valid conclusion, but this usually isn't the point of the test—you want to get a sense for what people like and don't like about your game, not find a statistically significant result.

Find **three** people who are willing to play your game. It is preferable for them to not be in this class (and thus not know anything about what you are testing). Ask each of your participants to play against both versions of the AI. **Do not tell them which AI is which before they play, and do not give them any information about how the AIs work.** Have a notebook with you while you watch them play.

For each play session, take notes while you observe your participants playing the game. Do they seem to be struggling? Do they seem frustrated? Is there behavior you see in the ghosts that seems unusual, broken, too easy, or too difficult? Make sure you take note of which AI the player is playing against when you see these behaviors. After the player has played against both AIs, ask them what they think about the two AIs. Did one seem easier to play against? Was one more fun to play against? Were there any moments during their play that were especially memorable? Write down what you ask them and how they answer.

When you have finished your playtest, write up your notes into a short report. Make sure to mention not only what your participants reported, but also *why* you think they responded to each AI the way they did.

ASSIGNMENT WRITEUP

As noted above, you must create a writeup to accompany your code. This writeup **must** include:

- A paragraph that notes any times that you deviated from the original ghost specification and states why you needed to do so.

- A two paragraph reflection on how what you learned during this assignment relates to what we have discussed in class, or what you have learned in other contexts (projects, internships, personal interests, etc.)
- Your playtesting report.

Please start working on this early! If you wait until the last minute to start making your controller, you will not have enough time to complete the playtesting portion of the assignment.

EVALUATION

This assignment will be graded according to the following rubric. The controller is worth 60% of the grade, the playtest report is worth 30%, and code style is worth 10%. **In order to earn any credit, your code must compile.**

	Excellent (9-10)	Good (6-8)	Not Good (3-5)	Poor (0-2)
Controller	All ghosts perform as expected (within the limitations of the framework)	Most ghosts perform as expected, but there are some bugs	Ghost AI is not implemented to target pacman; ghosts do not work as expected but there is evidence of an attempt	Little to no evidence of effort to make the ghosts work
Playtest Report *though we do not grade reports based on grammar, please make every effort to present a professional and clear report*	Well-written report that provides both the data from the playtests and analysis/opinion about why the design of the system led to results	Well-written report that recounts data from the playtests, but does not provide adequate reflection.	Playtest was not performed as required (e.g. too few participants, or does not compare AI systems); does not report on what was required in the playtest sessions	No evidence that playtesting occurred, or playtesting was carried out entirely incorrectly
Code Style	Code is well-commented; code is formatted clearly and legible (e.g. appropriate variable names); good code re-use (if appropriate); controller classes and any supporting code are placed in the required package	Some deficiencies in style, but overall code is still legible (i.e. does not mean all of the requirements for "excellent" but does meet many of them)	Very few comments; poorly chosen variable names, lack of code re-use	Completely illegible code; lack of comments; very poor coding style

EXTRA CREDIT

For a maximum of 10 extra credit percentage points, you may choose to implement your own custom controller, and compare it to the controller you implement that mimics the original pacman AI. If you choose to do this, use this custom controller in your playtest instead of the random ghost controller. You should also include a writeup (max. 1 page) of how the controller works and what your goals were for it. **Extra credit points will be awarded at the discretion of the instructional staff**, and will be based on some combination of sophistication of implementation, underlying theories related to design and player experience, and quality of player experience compared to the original AI.

SUBMISSION INSTRUCTIONS

Turn in a .zip file containing:

- Your .java file for the controller that mimics the original Pacman AI.
- (Optional) Your .java file for your extra credit controller.
- Your readme file (.txt)
- Your project writeup, including the playtest report (.pdf)

Assignments must be turned in via InstructAssist. **Emailed assignments will not be accepted.**